



# Regularized decomposition of large scale block-structured robust optimization problems

Wim van Ackooij, Nicolas Lebbe, Jérôme Malick

## ► To cite this version:

Wim van Ackooij, Nicolas Lebbe, Jérôme Malick. Regularized decomposition of large scale block-structured robust optimization problems. 2016. hal-01492908

**HAL Id: hal-01492908**

**<https://hal.science/hal-01492908>**

Preprint submitted on 22 Mar 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Regularized decomposition of large scale block-structured robust optimization problems

Wim van Ackooij · Nicolas Lebbe · Jérôme Malick

Received: date / Accepted: date

**Abstract** We consider a general robust block-structured optimization problem, coming from applications in network and energy optimization. We propose and study an iterative cutting-plane algorithm, generic for a large class of uncertainty sets, able to tackle large-scale instances by leveraging on their specific structure. This algorithm combines known techniques (cutting-planes, proximal stabilizations, efficient heuristics, warm-started bundle methods) in an original way for better practical efficiency. We provide a theoretical analysis of the algorithm and connections to existing literature. We present numerical illustrations on real-life problems of electricity generation under uncertainty. These clearly show the advantage of the proposed regularized algorithm over classic cutting plane approaches. We therefore advocate that regularized cutting plane methods deserve more attention in robust optimization.

**Keywords** Large scale block-structured problems · robust optimization · cutting-plane methods · bundle methods · unit-commitment.

**Mathematics Subject Classification (2000)** MSC 90C15 · MSC 90C25 · MSC 49M27

## 1 Introduction

### 1.1 Robust block-structured optimization

We consider block-structured optimization problems of the following form in  $\mathbb{R}^n$  (with  $n = n_1 + \dots + n_m$ )

$$\begin{aligned} \min_{x=(x_1, \dots, x_m)} \quad & \sum_{i=1}^m f_i(x_i) + \psi(x) \\ \text{such that } & x_i \in X_i, \quad \text{for all } i = 1, \dots, m, \end{aligned} \tag{1}$$

---

W. van Ackooij  
EDF R&D, OSIRIS,  
7 Boulevard Gaspard Monge, 91120 Palaiseau, France E-mail: wim.van-ackooij@edf.fr

N. Lebbe  
CEA LETI  
17 Avenue des Martyrs, 38000 Grenoble, France E-mail: lebbe.nicolas@gmail.com

J. Malick  
CNRS, LJK  
Bâtiment IMAG, Université de Grenoble, 38000 Grenoble, France E-mail: jerome.malick@univ-grenoble-alpes.fr

where  $\psi : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  and  $f_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}$  are given convex mappings and  $X_i \subseteq \mathbb{R}^{n_i}$  arbitrary compact sets. The complicating function  $\psi$  has a coupling effect on the specific block-structure: whenever  $\psi \equiv 0$ , problem (1) can in principle be solved by solving the  $m$  smaller problems

$$\min_{x_i \in X_i} f_i(x_i). \quad (2)$$

Such block-structured optimization problems appear in energy optimization (see e.g., [50] on unit-commitment problems) and in network optimization (see e.g., [15] with applications in network utility maximization and linear network flow problems).

We now consider that the coupling function  $\psi$  is further subject to uncertainty, and that we want to take optimal decisions over the whole uncertainty set. More precisely we assume that  $\psi$  depends on a parameter  $d$  lying in a given uncertainty set  $\mathcal{D} \subset \mathbb{R}^p$  which leads to the robust optimization problem

$$\begin{aligned} \min_{x=(x_1, \dots, x_m)} \quad & \sum_{i=1}^m f_i(x_i) + \sup_{d \in \mathcal{D}} \psi(x, d) \\ \text{such that} \quad & x_i \in X_i, \quad \text{for all } i = 1, \dots, m \end{aligned} \quad (3)$$

where the function  $\psi : \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R} \cup \{+\infty\}$  is convex with respect to the first variable. In section 2, we give several examples of situations where such robust block-structured optimization problems appear. Note also that (3) can have an interpretation in the terminology of stochastic optimization with  $\psi$  viewed as a full second stage (e.g., [68, eq. (1)]), and as a Benders formulation (e.g., [26]). The goal of this paper is to design a generic efficient method for solving this robust optimization problem in large scale context by leveraging on its specific block-structure.

## 1.2 Models of uncertainty, reformulation, and cutting-planes

Formally, the robust optimization problem (3) has the same structure as the initial problem (1) by just replacing  $\psi$  with the convex function  $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$  defined as

$$\Psi(x) := \sup_{d \in \mathcal{D}} \psi(x, d). \quad (4)$$

However, the practical difficulty of solving the two problems (3) and (1) strongly depends on the model of uncertainty  $\mathcal{D}$ . In some particular cases, the two problems have comparable computational costs. For example, when the set  $\mathcal{D}$  is a finite set with small cardinality, computing  $\Psi$  in (4) consists of computing  $\psi(x, d)$  finitely many times. This is the common situation in stochastic programming and does not necessarily cause modelling issues if we allow the set of scenarios  $\mathcal{D}$  to be sufficiently representative within a large expressive set of “realizations” (typically called scenarios). It is possible indeed to select representative scenarios by quantization (see e.g., [45]), methods from scenario reduction (see e.g., [31]), or adaptive partitioning schemes (see e.g., [49, 56]). These methods are well understood for objective functions involving expectations, but not for worst case objective functions. Their applications in our robust optimization context is unclear as it would raise many theoretical questions. These questions are related to how to produce a set  $\mathcal{D}'$ , close to  $\mathcal{D}$ , but significantly smaller than  $\mathcal{D}$  or such that (4) is cheaper to compute. This raises questions on distances between the optimal solutions (and associated optimal solution sets) in this context, which seems to be a new research area, e.g., [29].

For better modelling of uncertainty in our robust optimization situation, we would like to use directly richer uncertainty sets  $\mathcal{D}$ , such as finite sets with an exponential number of scenarios (see e.g., [43]) or elementary geometrical shapes (see e.g., [4, 6, 27, 28]) as

$$\mathcal{D} = \left\{ d \in \mathbb{R}^T : \|M(d - \bar{D})\|_p \leq \kappa \right\}, \quad (5)$$

where  $p \geq 1$  refers to a choice of norm,  $M$  is an appropriate matrix and  $\bar{D}$  the “center” of the shape. This shape includes famous models as hyper-rectangles ( $p = \infty$  and  $M = I$ ) and hyper-ellipsoids ( $p = 2$ ). For these choices of uncertainty in general, solving the robust problem (3) is much more complicated than solving (1), because computing  $\Psi$  is usually expensive (and can be NP-hard). We refer the reader to [5, 8] for general presentation and to [43] for an extensive discussion on NP-hardness and tractability of the computation of  $\Psi$ . For a concrete construction of  $\mathcal{D}$  in another way, we refer the reader to [1].

For rich and complicated sets  $\mathcal{D}$ , it may be difficult to reformulate the problem into a tractable form. In general the robust counterpart to an arbitrary convex optimization problem is intractable, see e.g., [5] (even if some approximation results for robust convex problems with a conic structure are discussed in [11]). However we can always trivially generate a linearization or a cutting-plane which lies below the function  $\Psi$ . Indeed, for any given arbitrary  $d \in \mathcal{D}$ , convexity of  $\psi(\cdot, d)$  yields

$$\psi(x, d) + \langle g, z - x \rangle \leq \psi(z, d) \leq \Psi(z), \quad \text{for any } g \in \partial_x \psi(x, d), \quad (6)$$

so we get a cutting plane of  $\Psi$  given by a value  $\Psi_x = \psi(x, d)$  and a vector  $g_x = g$ . Here  $\partial_x \psi(x, d)$  denotes the sub-differential of convex analysis of the mapping  $x \mapsto \psi(x, d)$ . In the terminology of nonsmooth optimization (see e.g., [42, 44]), we have a so-called (uncontrolled) inexact oracle. When  $d$  maximizes  $\psi(x, \cdot)$  over  $\mathcal{D}$  (that is  $\Psi_x = \psi(x, d) = \Psi(x)$ ), this linearization touches the graph of the function  $\Psi$ , and the oracle is said to be exact. Calling this oracle at several points  $x^1, \dots, x^k$  in the domain of  $\Psi$ , we can set up the so-called cutting-plane model:

$$\check{\Psi}_k(x) := \max_{j=1, \dots, k} \{ \Psi_{x^j} + \langle g_{x^j}, x - x^j \rangle \} \leq \Psi(x), \quad (7)$$

lying below the function by convexity. Such models are often used in robust optimization by cutting-plane algorithms; see e.g., the recent article [9] for a numerical comparison in robust optimization between the two main approaches (reformulation to a deterministic problem vs iterative cutting plane method). In our context with general uncertainty set  $\mathcal{D}$ , cutting plane algorithms should be methods of choice. However in a direct application, we cannot exploit the block-structure of the problem. This prevents the conventional use of cutting-plane methods in the large-scale setting of this paper, because the master optimization problem can not be handled directly. This is developed in section 3.

### 1.3 Contributions and outline of the article

This paper presents an efficient iterative algorithm for solving large scale robust optimization problem (3), which has the three following features:

- it is generic for any uncertainty set  $\mathcal{D}$ ,
- it exploits the block-structure of (3),
- it incorporates numerical improvement (stabilization, warm-starting...) to tackle large-scale instances.

The only assumption on  $\psi$  and  $\mathcal{D}$  is that we can (approximately) solve (4). Thus cutting-plane approximations of  $\Psi$  can be built using (6) and (7). A straightforward use of an iterative cutting-plane algorithm would make the “master” problems computing the next iterates hard to solve, due to their large-scale. The first algorithmic specificity of our approach is to leverage on the block-structure of (3) by decomposing the master problem by duality into parallel subproblems of the form (2). We make no specific structural assumptions on the compact sets  $X_i$  which can be “hard to handle” (i.e., a set with nonlinearities, nonconvexities, mixed-integer variables or probabilistic constraints). The practical assumption is that solving (2) is a relatively “easy” task compared to solving (1) or (3).

The second key algorithmic feature of the algorithm is the quadratic regularization of the cutting-plane approach. Regularized cutting-plane methods have proved to be highly efficient in nonsmooth optimization [2, 12], stochastic optimization (see e.g., [21] and [69]), chance constrained optimization (e.g., [55, 61]), energy management ([24]) and combinatorial optimization [13]. However they have not received much

attention in the context of robust optimization. In contrast, the recent article [9] reports that stabilizations of cutting-plane methods “did not yield significant improvements in computation time in our experiments” on a variety of problem instances for both robust linear optimization and robust mixed-integer optimization problems using polyhedral and ellipsoidal uncertainty sets. We will show here the exact opposite: our numerical experiments on real-life robust problems in energy optimization show a clear domination of the regularized cutting-plane method.

We design here a (regularized) cutting-plane algorithm adapted to (3), inspired but different from recent developments in nonsmooth optimization (see e.g., [44] and [30]). The main difference is that our master problem cannot be solved directly, which has practical implications (including the use of a second layer of cutting-plane) and theoretical impacts (including the fact that the analysis is not covered by recent studies of inexact bundle methods). We then provide an original analysis of the convergence and of the quality of the obtained final solution.

The paper is structured as follows. In section 2, we discuss how the structure of problem (3) encompasses several problems already considered in the literature. We also mention our motivating example and some of its features arising from unit-commitment. The cutting-plane approach designed for (3) is presented in Section 3, together with the ingredients for practical efficiency and a theoretical study. Numerical experiments are presented in section 4, showing the advantage of regularization over the simpler variant. The conclusion summarizes this work and advocates that regularized cutting-plane algorithms deserve more attention in robust optimization. The paper ends with appendices to recall notions and terminology used in this paper of nonsmooth optimization and operations research.

## 2 Examples of robust block-structured problem

We present in this section several situations where the robust two-stage optimization problem (3) appears. We partially focus on the robust unit-commitment problem that we use in our numerical illustrations. We also briefly list other applications in stochastic programming.

### 2.1 Motivating application: large scale unit-commitment

Unit-commitment is one of the basic problems in energy optimization; see the recent review [50]. We assume that there are  $m$  units (thermal and hydro valleys) over the time interval  $\{1, \dots, T\}$ . Each individual unit has its own cost  $f_i: \mathbb{R}^{n_i} \rightarrow \mathbb{R}$  and production constraints  $X_i \subset \mathbb{R}^{n_i}$ . We consider an abstract form of this problem with the decision vector  $x = (x_1, \dots, x_m)$

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \sum_{i=1}^m f_i(x_i), \\ \text{such that} \quad & x \in X \cap C. \end{aligned} \tag{8}$$

Here  $X$  is the set of technically feasible production schedules,  $n = \sum_{i=1}^m n_i$  and  $C$  the set of system wide constraints. We consider, for instance, the power balance constraints, as the set of system-wide constraints, that is:

$$C = \{x \in \mathbb{R}^n : Ax \geq d\}, \tag{9}$$

where  $d \in \mathbb{R}^T$  is the (net) customer/system load and  $Ax$  the total amount of produced power. We remark that realistic modelling of the French electrical system gives birth to a large scale unit-commitment problem of this form, e.g., the example given in the numerical section.

When renewable generation has overall high installed capacity, uncertainty should be accounted for in the model. We choose to consider an uncertain customer load and we further assume that complete recourse decisions exist with respect to market conditions (following [43]). Typically the commitment variables,

i.e., the integer variables, are restricted to the first stage and assumed to be fixed ahead of time, unable to respond to changes in  $d$ . A second stage deals with solving an “economic dispatch problem”. The latter problem deals with finding appropriate power output levels of power plants in order to meet the actual customer load. We consider a general piecewise affine penalization function  $\psi_t : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  given by

$$\psi_t(y, d) = \max_{i=1, \dots, n_t} \{a_{i,t}(d - y) + b_{i,t}\}. \quad (10)$$

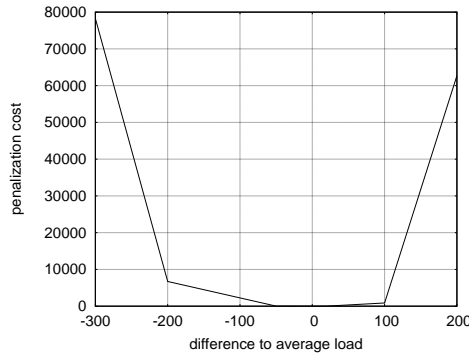
for coefficients  $a_{i,t}, b_{i,t}$  (e.g., see Figure 1 below). This suggestion covers for instance the penalization function of [23, Fig 2.2]. It also covers the situation considered in [3] and [43] wherein

$$\psi_t(y, d) = \max \{\alpha(d - y), \beta(d - y)\}. \quad (11)$$

With this model, we further consider the worst case outcomes, so we are interested in solving the optimization problem (3) with this penalization function

$$\Psi(x) = \sup_{d \in \mathcal{D}} \sum_{t=1}^T \psi_t(d_t, (Ax)_t), \quad (12)$$

where  $(Ax)_t$  refers to total generation at instant  $t$ .



**Fig. 1** Example of a penalization function  $\psi_t$

## 2.2 Other examples

*Adaptive robust optimization* A popular set of approaches for a block-structured problem (3) is based on 2-stage robust optimization, also called adaptive robust optimization; see for example its application to unit-commitment in [10]. In this situation the mapping  $\psi(x, d)$  is of the following form:

$$\psi(x, d) = \begin{cases} \min_{y \in \mathbb{R}^p} & \langle b(d), y \rangle \\ \text{such that} & A(d)x + B(d)y \leq h(d) \\ & y \in Y, \end{cases} \quad (13)$$

where  $Y$  can be an arbitrary convex set and  $A(d), B(d), h(d), b(d)$  are vectors and matrices of appropriate dimension depending on  $d$ .

*Worst case penalization* When a coupling constraint of the type  $g(x) \leq b$  is present in (1), an approach is to penalize this constraint with a mapping attributing a value  $\varphi(g(x) - b)$  to the violation of the constraint; see e.g., [43]. The mapping  $\varphi$  should be increasing and satisfy  $\varphi(z) = 0$  for  $z \leq 0$ . Then when realizing that  $g$  and/or  $b$  are subject to uncertainty, this leads to the situation described above with  $\psi(x, d) := \varphi(g_d(x) - b(d))$ . This is also a special case of simple recourse since

$$\psi(x, d) = \min_z \{\varphi(z) \text{ s.t. } g_d(x) - b(d) \leq z\}. \quad (14)$$

*Semi-infinite, Soyster like problems* Problem (1) can involve constraints of the type

$$A(d)x \leq b(d) \quad \text{for all } d \in \mathcal{D}, \quad (15)$$

as in [9]. This case fits into the general framework by defining

$$\psi(x, d) = \min_{y \leq 0} \{ \langle 0, y \rangle \text{ s.t. } A(d)x + y \leq b(d) \}.$$

The value function is then used to define feasibility cuts.

Whenever  $x \notin \text{dom } \Psi(x)$ , one can typically also obtain some first order information: a feasibility cut. Then by solving the auxiliary problems  $\psi^f(x, d) := \{ \min_y \|y\| : \text{s.t. } A(d)x + y \leq b(d) \}$ , it becomes clear that  $\text{dom } \Psi = \bigcap_{d \in \mathcal{D}} \{x \in \mathbb{R}^n : \psi^f(x, d) \leq 0\}$ . Moreover  $\psi^f(\cdot, d)$  is a convex mapping, and an optimal dual multiplier to the equation  $A(d)x + y \leq b(d)$ , satisfies  $-A(d)^\top \lambda \in \partial \psi^f(x, d)$ . From this discussion it becomes clear that feasibility cuts are but another cutting-plane model for taking into account a convex constraint of the type  $\Psi^f(x) \leq 0$ , where  $\Psi^f(x) = \sup_{d \in \mathcal{D}} \psi^f(x, d)$ .

*Finite small cardinality  $\mathcal{D}$*  This situation considered in [60] and [53] has a mapping that itself is subject to complicating non-convex and/or combinatorial constraints. In this situation  $\psi(x, d)$  can be written as:

$$\psi(x, d) = \begin{cases} \min_{(y_1, \dots, y_m)} & \sum_{i=1}^m g_i(y_i) \\ \text{such that} & \sum_{i=1}^m \theta_i(y_i) \leq b(d) \\ & y_i \in Y_i \\ & Px = Py, \end{cases} \quad (16)$$

Although  $\psi(x, d)$  is not convex in  $x$  because of the potentially non-convex structure of some of the  $Y_i$ , an implicitly computable convexification is available using the Lagrangian dual of (16) with respect to the coupling constraints  $\sum_{i=1}^m \theta_i(y_i) \leq b(d)$  and  $Px = Py$ . An extensive discussion of the convexification effect of such a scheme as well as an efficient algorithm for dealing with these problems is discussed in a similar case in [60].

### 3 (Regularized) cutting-plane methods

In this section we present two cutting-plane algorithms for solving the block-structured robust problems (3). We start in section 3.1 with recalling the cutting-plane approximation in general and how to deal with them by duality for our block-structured problems. In section 3.2 we discuss an elementary cutting-plane method for solving (3) as it helps to show the ideas and to formalize the notation. Then we present in section 3.3 the regularized cutting-plane method, that we use for the numerical illustrations of the next section. We use the classical terminology in convex optimization and operation research (see the foundational references [7, 26, 32, 33], and the discussion in appendices A and B if necessary).

#### 3.1 Cutting-plane models

Cutting-plane methods are one of the major approaches in robust optimization, and more generally in operation research. As discussed in the introduction, we have, in our situation, an oracle producing, for a given  $x$ :

- either an optimality cut: a lower linearization of  $\Psi$  (if  $x \in \text{dom } \Psi$ ),
- or a feasibility cut: a cutting-plane separating  $x$  and  $\text{dom } \Psi$  (if  $x \notin \text{dom } \Psi$ ).

After calling the oracle on several iterates  $x^1, \dots, x^k$ , we denote with  $\mathcal{O}_k$  the set of iterates at which an optimality cut was derived and  $\mathcal{F}_k$  the set of iterates at which a feasibility cut was obtained. Then we define the optimality cutting-plane model  $\check{\Psi}_k$  as (7) with respect to the set of iterates  $\mathcal{O}_k$  only; and similarly we introduce the feasibility cutting-plane model  $\check{\Psi}_k^f$  defined with respect to  $\mathcal{F}_k$  only.

The idea of the cutting-plane method is to use the above-defined cutting-plane models to replace the unknown  $\Psi$  and  $\text{dom}\Psi$  in problem (3). More specifically, we replace the original convex objective  $\Psi$  by the convex piecewise affine function  $\check{\Psi}_k$  and the convex implicit constraint  $x \in \text{dom}\Psi$  by the convex polyhedral constraint  $\check{\Psi}_k^f(\cdot) \leq 0$ . Thus we consider the following approximation of problem (3)

$$\begin{aligned} \min_{(x,r) \in \mathbb{R}^{n+1}} \quad & \sum_{i=1}^m f_i(x_i) + r \\ \text{such that} \quad & x \in X, \quad \check{\Psi}_k(x) \leq r, \quad \check{\Psi}_k^f(x) \leq 0, \end{aligned} \quad (17)$$

which is again equivalent to:

$$\begin{aligned} \min_{(x,r) \in \mathbb{R}^{n+1}} \quad & \sum_{i=1}^m f_i(x_i) + r \\ \text{such that} \quad & x \in X, \\ & \Psi(x^\ell) + \langle g^\ell, x - x^\ell \rangle \leq r, \quad \ell \in \mathcal{O}_k \\ & \Psi_k^f(x^\ell) + \langle g_\ell^f, x - x^\ell \rangle \leq 0, \quad \ell \in \mathcal{F}_k. \end{aligned}$$

This problem, called master problem at iteration  $k$ , is still a difficult optimization problem because of the large scale and potential arbitrarily bad structure of the compact sets  $X_i$ . To leverage on the block-structure of this problem, our idea is to move to the Lagrangian dual with respect to the coupling constraints. Note that from a dual view, the distinction between feasibility and optimality cuts becomes anecdotic. For all  $k \geq 1$ ,  $|\mathcal{O}_k| + |\mathcal{F}_k| = k$ , let us define:

$$(\bar{\Psi}(x^\ell), \bar{g}^\ell) := \begin{cases} (\Psi(x^\ell), g^\ell) & \text{if } \ell \in \mathcal{O}_k \\ (\Psi^f(x^\ell), g_\ell^f) & \text{if } \ell \in \mathcal{F}_k \end{cases} \quad (18)$$

**Lemma 1 (Oracle for the  $k$ -th dual function)** *The concave dual function  $\Theta_k$  associated with (17) can be expressed, for all  $\mu \in \mathbb{R}_+^k$  such that  $\sum_{\ell \in \mathcal{O}_k} \mu_\ell = 1$ , as*

$$\Theta_k(\mu) := \min_{(x_1, \dots, x_m) \in \prod_{i=1}^m X_i} \sum_{i=1}^m f_i(x_i) + \left\langle \sum_{\ell=1}^k \mu_\ell \bar{g}^\ell, x \right\rangle + \left( \sum_{\ell=1}^k \mu_\ell (\bar{\Psi}(x^\ell) - \langle \bar{g}^\ell, x^\ell \rangle) \right). \quad (19)$$

Thus solving the  $m$  subproblems

$$\min_{x_i \in X_i} f_i(x_i) + \left\langle \sum_{\ell=1}^k \mu_\ell \bar{g}_i^\ell, x_i \right\rangle$$

gives the value  $\Theta_k(\mu)$ , a sub-gradient

$$(\bar{\Psi}(x^1) + \langle \bar{g}^1, x(\mu) - x^1 \rangle, \dots, \bar{\Psi}(x^k) + \langle \bar{g}^k, x(\mu) - x^k \rangle)^\top \in \partial \Theta_k(\mu)$$

and an associated primal solution  $x(\mu) = (x_1(\mu), \dots, x_m(\mu)) \in X$ .

*Proof* Starting from (17) we introduce a Lagrangian multiplier  $\mu \in \mathbb{R}_+^k$  and define the dual function by

$$\Theta_k(\mu) := \min_{\substack{x_i \in X_i \\ r \in \mathbb{R}}} \sum_{i=1}^m f_i(x_i) + r + \left( \sum_{\ell=1}^k \mu_\ell (\bar{\Psi}(x^\ell) + \langle \bar{g}^\ell, x - x^\ell \rangle) \right) - \sum_{\ell \in \mathcal{O}_k} \mu_\ell r.$$



As soon as we choose  $r \neq 0$  and any  $x_i$ , we can pick  $\mu = (t, 0, \dots, 0)$  with  $|t| \rightarrow +\infty$  which leads to  $\Theta_k(\mu) \rightarrow -\infty$ . Thereby we see that we can restrict  $\mu$  to the subset of  $r$  on which we have  $r = \sum_{\ell \in \mathcal{O}_k} \mu_\ell r$ , which is exactly the set  $\{\mu : \sum_{\ell \in \mathcal{O}_k} \mu_\ell = 1\}$ . Using this restriction the variable  $r$  vanishes and we find equation (19) for the dual function. Finally, applying a standard result of Lagrangian duality (see e.g., [33, Prop. XII.2.2.2]), the remaining assertions follow.  $\square$

This lemma shows that we can leverage on the block-structure of the problem to set up an efficient oracle of the dual function associated with (17) by decomposing the computation into  $m$  smaller subproblems. To maximize this concave dual function, the so-called bundle methods (see e.g., [33, 44]) are then the methods of choice. We will use these methods as “inner engines” within the algorithms presented in the next two sections. For the convenience of non-expert readers, we make a short presentation of proximal bundle methods in appendix A.

### 3.2 Cutting-plane method

We propose to solve (3) with a standard cutting-plane algorithm where the novelty is to use a duality to exploit the block-structure of the problem. From the developments of the previous section, this approach gives the following algorithm (inspired from the cutting-plane method of [60] proposed in another context).

- (Step 0 : Initialization) : Generate several initial points in  $X$  to build an initial model for  $\Psi$ . Set  $k$  and the stopping tolerance  $\delta_{\text{stop}}$ .
- (Step 1 : Lagrangian dual) : Use a bundle method to maximize the dual function (19) with the additional constraint  $\sum_{\ell \in \mathcal{O}_k} \mu_\ell = 1$ . Get an approximate dual solution  $\mu^{k+1}$ .
- (Step 2 : Primal Recovery) : Exploit the dual information generated in Step 1 to recover a solution of good quality  $x^{k+1} \in X$ . Estimate the duality gap  $\Delta_k^A := f(x^{k+1}) + \check{\Psi}_k(x^{k+1}) - \Theta_k(\mu^{k+1})$ .
- (Step 3 : Oracle call) : Call the oracle for  $\Psi$  at  $x^{k+1}$  to enrichen the model for  $\Psi$  by adding a (feasibility or optimality) cut.
- (Step 4 : Stopping test) : In the case where  $x^{k+1}$  lies in  $\text{dom } \Psi$ , stop the algorithm if

$$\Psi(x^{k+1}) - \check{\Psi}_k(x^{k+1}) \leq \delta_{\text{stop}}. \quad (20)$$

Otherwise let  $k = k + 1$  and return in Step 1.

This algorithm uses a standard cutting-plane machinery with two specific working horses: the bundle method of Step 1 and the primal recovery of Step 2. We discuss these key computational ingredients later in section 3.4 and study here the convergence of the overall algorithm.

We emphasize that the accuracy of the final iterate, though estimated, is not controlled explicitly by the algorithm due to the potential non-convexities in  $X$ . The meaning of the stopping test is thus the following.

**Lemma 2 (Approximate optimality)** *If at iteration  $k$  the cutting-plane algorithm terminates then  $x^{k+1}$  is an  $(\Delta_k^A + \delta_{\text{stop}})$ -optimal solution to problem (3).*

*Proof* Letting  $x^* \in X^1$  be the optimal solution to problem (3), we derive

$$\begin{aligned} f(x^*) + \Psi(x^*) &\geq f(x^*) + \check{\Psi}_k(x^*) \geq f(x^{k+1}) + \check{\Psi}_k(x^{k+1}) - \Delta_k^A \\ &\geq f(x^{k+1}) + \Psi(x^{k+1}) - \Delta_k^A - \delta_{\text{stop}}, \end{aligned}$$

when using convexity of  $\Psi$ ,  $\Delta_k^A$  optimality of  $x^{k+1}$  for (17), and the stopping criteria respectively.  $\square$

We then have the following convergence result under two assumptions on finite feasibility iterations and boundedness of subgradients. These assumptions are due to the fact that  $\Psi$  might not be defined everywhere and they are standard in similar contexts (for example, they are already present in the seminal work [26] on Benders decomposition).

**Proposition 1 (Convergence of the cutting plane method)** *Assume that the algorithm does not generate infinitely many consecutive feasibility cuts, that all the used subgradients of  $\Psi$  are uniformly bounded. Let  $\delta_{\text{stop}} > 0$ , then the algorithm terminates for some finite  $k$  and  $x^{k+1}$  is a  $(\Delta_k^A + \delta_{\text{stop}})$ -optimal solution to problem (3).*

*Proof* Let us establish convergence by contradiction; we assume that there is a  $\delta_{\text{stop}} > 0$  such that for all  $k$  either  $x^{k+1} \notin \text{dom } \Psi$  or

$$\Psi(x^{k+1}) - \check{\Psi}_k(x^{k+1}) > \delta_{\text{stop}}.$$

Let us denote by  $K$  the set of indexes of the iterations where the above inequality is satisfied. This set is infinite by assumption. For any  $k \geq 1$  and  $\ell \leq k$ , we have by (7),

$$\delta_{\text{stop}} < \Psi(x^{k+1}) - \check{\Psi}_k(x^{k+1}) \leq \Psi(x^{k+1}) - \Psi(x^\ell) - \langle g^\ell, x^{k+1} - x^\ell \rangle.$$

Using the Cauchy-Schwarz inequality yields:

$$\delta_{\text{stop}} < |\Psi(x^{k+1}) - \Psi(x^\ell) - \langle g^\ell, x^{k+1} - x^\ell \rangle| < |\Psi(x^{k+1}) - \Psi(x^\ell)| + \|g^\ell\| \|x^{k+1} - x^\ell\|. \quad (21)$$

By assumption, all the  $\|g^\ell\|$  are bounded by a certain  $C \geq 0$  and by compactness of  $X$  the sequence  $(x^k)_k$  admits a cluster point. Therefore, using also the fact that  $\Psi$  is continuous, the right hand side in (21) tends to 0 along an appropriate subsequence, yet this immediately leads to a contradiction. Lemma 2 allows us to conclude.  $\square$

In theory the boundedness of the subgradients can be ensured on the interior of the domain of  $\Psi$ , but not on its boundary. In practice this is not a problem, since computer codes naturally require “bounded” subgradients in order to ensure well-posedness of the linear systems involved in the solution process. When  $\text{dom } \Psi = \mathbb{R}^n$  as in our application, the convergence is guaranteed, as formalized in the following easy corollary.

**Corollary 1** *When  $\text{dom } \Psi = \mathbb{R}^n$ , the assumptions of Proposition 1 hold.*

*Proof* The set  $X$  is compact, and therefore so is  $\text{Co}(X)$  by [32, III.1.4.3], where  $\text{Co}(X)$  denotes the convex hull. Moreover,  $\Psi$  is convex on  $\text{Co}(X)$  so in particular locally Lipschitz continuous at each  $x \in \text{Co}(X)$ . By compactness,  $\Psi$  is Lipschitz continuous on all  $\text{Co}(X)$  and using [16, proposition 2.1.2] there is a constant  $C > 0$  such that  $\|g^\ell\| \leq C$  for all  $\ell \in \{1, \dots, L\}$  because  $g^\ell \in \partial \Psi(x^\ell)$  with  $x^\ell \in \text{Co}(X)$ .  $\square$

We finish this section by two remarks. First the above convergence results provides the information that only the final precision matters: in early iterations one could therefore solve (17) with a very imprecise method, as long as, when nearing convergence, problem (17) is solved with more precision. Second, although our interest originates from problems wherein (3) is large scale, the above algorithm can also be employed for smaller scale problems. For some problems, it might be possible to short-cut the bundle methods of Step 1 and solve directly (17). Proposition 1 and the preceding remark on imprecise solution still hold in this case.

### 3.3 Regularized cutting plane method

The cutting plane method in practice shows the well known oscillation effect and slow convergence, as illustrated in the numerical experiments of the next section (see Figure 3). To improve efficiency of the cutting plane method, we use the standard technique for adding proximal stabilization (see e.g., the textbook [33]), that usually significantly reduces the overall computational effort.

The regularized cutting-plane method uses the same approach and the same numerical components as for the previous algorithm. The main difference is that we add a "proximal" quadratic regularization term to the master problem. Instead of solving (17), we solve

$$\begin{aligned} \min_{(x_1, \dots, x_m), r} \quad & \sum_{i=1}^m f_i(x_i) + r + \frac{1}{2t_k} \|x - \hat{x}^k\|_2^2 \\ \text{such that } \quad & x \in X, \quad \check{\Psi}_k(x) \leq r, \quad \check{\Psi}_k^f(x) \leq 0, \end{aligned} \quad (22)$$

where  $\hat{x}_k \in X$  is a current 'best' feasible iterate. Notice that other types of quadratic regularizations of cutting-plane algorithms could be considered, including stabilization by levels or trust-regions; see e.g., [41]. Proximal stabilizations that we use here have not been considered yet in a context of the Benders decomposition, contrary to level and trust-region stabilizations (see [58, 66]). However, the proximal stabilization is the most popular in operation research in general [13].

As in the previous section, an efficient version of a (proximal) regularized cutting-plane algorithm has to reveal the block-structure of the master problem (22). Despite the quadratic term, the dual of problem (22) still preserves its decomposability, as formalized in the next lemma. The proof of the lemma follows the exact same lines as the one of Lemma 1 and is therefore omitted.

**Lemma 3 (Oracle for the  $k$ -th dual function)** *The dual function  $\Theta_k^R : [0, 1]^k \rightarrow \mathbb{R}$  is defined for all  $\mu \in \mathbb{R}^k$  such that  $\sum_{\ell \in \mathcal{O}_k} \mu_\ell = 1$  by*

$$\Theta_k^R(\mu) := \min_{x \in X} f(x) + \frac{1}{2t_k} \|x - \hat{x}^k\|_2^2 + \left\langle \sum_{\ell=1}^k \mu_\ell \bar{g}^\ell, x \right\rangle + \left( \sum_{\ell=1}^k \mu_\ell (\bar{\Psi}(x^\ell) - \langle \bar{g}^\ell, x^\ell \rangle) \right). \quad (23)$$

Thus we have to solve the  $m$  subproblems

$$\min_{x_i \in X_i} f_i(x_i) + \left\langle \sum_{\ell=1}^k \mu_\ell \bar{g}_i^\ell - \frac{1}{t_k} \hat{x}_i^k, x_i \right\rangle + \left\langle x_i, \left( \frac{1}{2t_k} I \right) \right\rangle x_i,$$

which gives:

$$(\bar{\Psi}(x^1) + \langle \bar{g}_1^1, (x(\mu) - x^1) \rangle, \dots, \bar{\Psi}(x^k) + \langle \bar{g}_k^k, (x(\mu) - x^k) \rangle)^\top \in \partial \Theta_k^R(\mu)$$

This gives the following regularized cutting plane method:

- (Step 0 : Initialization) : Generate one or several initial elements  $x \in X$  and use these to build an initial model for  $\Psi$ . Set  $k = 1$ , the stopping tolerance  $\delta_{\text{stop}}$ ,  $t_1 > 0$ , and let  $\hat{x}^1 = x^1$  be one of these initial iterates.
- (Step 1 : Lagrangian dual) Use a bundle method to maximize (23) with the additional constraint  $\sum_{\ell \in \mathcal{O}_k} \mu_\ell = 1$ .
- (Step 2 : Primal Recovery) : Exploit the dual information generated in Step 1 to recover a solution of good quality  $x^{k+1} \in X$ . Estimate the duality gap  $\Delta_k^A := f(x^{k+1}) + \check{\Psi}_k(x^{k+1}) + \frac{1}{2t_k} \|x^{k+1} - \hat{x}^k\|_2^2 - \Theta_k^R(\mu^*)$ . Compute the predicted decrease  $\delta^k := f(\hat{x}^k) + \Psi(\hat{x}^k) - f(x^{k+1}) - \check{\Psi}_k(x^{k+1})$ .
- (Step 3 : Oracle call) : Call the oracle for  $\Psi$  to compute a value and subgradient at  $x^{k+1}$ , enriching the cutting-plane model by an optimality or a feasibility cut.

- (Step 4 : Descent condition) Check if the observed decrease is at least a fraction  $m > 0$  of predicted decrease  $\delta^k$ :

$$f(x^{k+1}) + \Psi(x^{k+1}) \leq f(\hat{x}^k) + \Psi(\hat{x}^k) - m\delta^k. \quad (24)$$

If (24) holds, let  $\hat{x}^{k+1} = x^{k+1}$ , else  $\hat{x}^{k+1} = \hat{x}^k$

- (Step 5 : Stopping test) : Check if

$$\delta^k \leq \delta_{\text{stop}} \quad (25)$$

and stop the algorithm if this holds. Let  $k = k + 1$ , choose  $t_{k+1}$  arbitrarily and return in Step 1.

This algorithm does not fit in the recent development of inexact nonsmooth optimization methods (see e.g., [20] and references therein) where the inexactness is the one of the oracle. Here the main inexactness comes from the fact that the approximating subproblem (17) is solved inexactly. The convergence of the algorithm is thus not covered by existing results. However, we can adapt standard rationale to get the following convergence result, under the same assumptions as the ones of Proposition 1.

**Proposition 2 (Convergence of regularized cutting plane method)** *Assume that the algorithm does not generate infinitely many consecutive feasibility cuts and that all used subgradients of  $\Psi$  are uniformly bounded. Let  $\delta_{\text{stop}} > 0$ , then the regularized algorithm terminates for some finite  $k$  and  $x^{k+1}$  is a  $(\Delta_k^A + \delta_{\text{stop}})$ -optimal solution to problem (3).*

*Proof* We prove this result by contradiction; we assume that

$$\delta^k = (f + \Psi)(\hat{x}^k) - (f + \check{\Psi}_k)(x^{k+1}) > \delta_{\text{stop}} \text{ for all } k \geq 1.$$

If there is an infinite number of iterations where (24) holds (called serious iterations, in the bundle terminology), then for all such iterations, we have

$$(f + \Psi)(\hat{x}^{k+1}) \leq (f + \Psi)(\hat{x}^k) - m\delta^k \leq (f + \Psi)(\hat{x}^k) - m\delta_{\text{stop}}.$$

Consequently, the fact that the objective function decreases infinitely many times by a fraction of at least  $m\delta_{\text{stop}}$  contradicts the fact that  $f + \Psi$  is bounded from below on the compact set  $X$ .

Consequently, there exists some  $\bar{k}$  such that  $\hat{x} := \hat{x}^k = \hat{x}^{\bar{k}}$  for all  $k \geq \bar{k}$ . By assumption there exist some  $C > 0$  such that  $\|g\| \leq C$  for all  $g \in \partial\Psi(x)$  and for all  $x \in X$ . Hence,

$$\begin{aligned} \delta_{\text{stop}} &\leq (f + \Psi)(\hat{x}) - (f + \check{\Psi}_k)(x^{k+1}) \\ &\leq (f + \Psi)(\hat{x}) - f(x^{k+1}) - \Psi(x^\ell) - \langle g^\ell, x^{k+1} - x^\ell \rangle \\ &\leq (f + \Psi)(\hat{x}) - f(x^{k+1}) - \Psi(x^\ell) + C \|x^{k+1} - x^\ell\| \end{aligned} \quad (26)$$

by (7) and the Cauchy-Schwarz inequality. The set  $X$  is compact, so by moving to a subsequence if needed, the sequence  $\{x^k\}_k$  can be assumed to admit a limit  $\bar{x}$ . Now (26) allows us to move to the  $\liminf$  and observe:

$$\begin{aligned} \delta_{\text{stop}} &\leq \liminf_{\ell \leq k, (\ell, k) \rightarrow \infty} [(f + \Psi)(\hat{x}) - f(x^{k+1}) - \Psi(x^\ell) + C \|x^{k+1} - x^\ell\|] \\ &= (f + \Psi)(\hat{x}) - \limsup_{\ell \leq k, (\ell, k) \rightarrow \infty} [f(x^{k+1}) - \Psi(x^\ell) + C \|x^{k+1} - x^\ell\|] \\ &= (f + \Psi)(\hat{x}) - (f + \Psi)(\bar{x}), \end{aligned} \quad (27)$$

by continuity of  $f$  and  $\Psi$  on  $\mathbb{R}^n$ . For each  $k \geq \bar{k}$ ,  $x^{k+1}$  is a null-iterate and therefore:

$$(f + \Psi)(\hat{x}) - (f + \Psi)(x^{k+1}) \leq m((f + \Psi)(\hat{x}) - (f + \check{\Psi}_k)(x^{k+1}))$$

Moving to the  $\liminf$  on both sides of the inequality, we establish:

$$(f + \Psi)(\hat{x}) - (f + \Psi)(\bar{x}) \leq m \left( (f + \Psi)(\hat{x}) - \limsup_k (f + \check{\Psi}_k)(x^{k+1}) \right),$$

which yields  $(1 - m)((f + \Psi)(\hat{x}) - (f + \Psi)(\bar{x})) \leq 0$  hence contradicting (27) since  $m \in (0, 1)$ .  $\square$

The contents of the discussion following Proposition 1 (about the assumptions, inexact resolution of the master in practice, and direct resolution of the master for small problems) still hold for this algorithm.

### 3.4 Algorithmic specificities

Two ingredients of the algorithms play a special role in the practical efficiency of the approach. First, the primal recovery heuristic (Step 2) should have a good balance between computational cost and quality of feasible solutions computed. Second, the bundle method for maximizing the dual of the approximate problem (Step 3) should not be a computational burden. We discuss these two points here.

#### 3.4.1 Primal recovery

Over the course of the dual bundle method, many elements of  $X$  are generated. A very simple primal recovery procedure is thus to simply retain the iterate that does best regarding the primal objective function. A more advanced primal recovery would use the convexity of certain subproblems (constrained by  $X_i$ ), as follows. Step 1 of the algorithm also provides simplicial multipliers (see e.g., [23], ) which can be used to form a convex combination of iterates. If  $X_i$  is convex, this combination will be feasible. An approach for recovery, called convexified best iterate, consists then in considering the best iterate after the convexification. Alternatively one can also solve the Dantzig-Wolfe-like optimization problem (e.g., [13])

$$\begin{aligned} \min_{z, x} \quad & \sum_{j=1}^p \sum_{i=1}^m z_i^j f_i(y_i^j) + \check{\Psi}_k(x) \\ \text{s.t.} \quad & x_i = \sum_{j=1}^p z_i^j y_i^j, i = 1, \dots, m, \quad \sum_{j=1}^p z_i^j = 1, i = 1, \dots, m \\ & \check{\Psi}_k^f(x) \leq 0, \quad z_i^j \in \{0, 1\}, i = 1, \dots, m, j = 1, \dots, p, \end{aligned} \quad (28)$$

where  $y^j$ ,  $j = 1, \dots, p$  are the solutions obtained while maximizing the Lagrangian dual. Note that, in the above problem (28),  $f_i(y_i^j)$  are fixed and  $x$  is an auxiliary variable, so that the problem (28) is a mixed-integer linear program for which efficient greedy heuristics exist. For example, this procedure was exploited in [51] in the context of unit-commitment.

#### 3.4.2 Warm-starting the master problem resolution

Maximizing the Lagrangian dual function (19) or (23) requires in turn building a cutting plane model of them. These cutting plane models are the essential building block of a bundle method (see e.g., the textbook [33] or appendix A). Since most of the computational cost resides in solving the  $m$  subproblems, it is useful to carry as much cutting-plane information as possible when moving from iteration  $k$  to iteration  $k + 1$ . This is the idea behind the following warm-starting procedure for the master problem.

Observe that we have two identifications. First, for any  $k \geq 1$  and any  $\mu \in \mathbb{R}^k$

$$\Theta_{k+1}((\mu, 0)) = \Theta_k(\mu).$$

Second, for any iterate  $\mu \in \mathbb{R}^k$  with associated  $x(\mu) \in X$  encountered while maximizing  $\Theta_k^R$ , we have

$$\begin{aligned} \Theta_{k+1}^R((\mu, 0)) &= f(x(\mu)) + \frac{1}{2t_{k+1}} \|x(\mu) - \hat{x}^{k+1}\|_2^2 + \left\langle \sum_{\ell=1}^k \mu_\ell g^\ell, x(\mu) \right\rangle + \left( \sum_{\ell=1}^k \mu_\ell (\Psi(x^\ell) - \langle g^\ell, x^\ell \rangle) \right) \\ &= \Theta_k^R(\mu) \end{aligned}$$

if  $t_{k+1} = t_k$  and  $\hat{x}^{k+1} = \hat{x}^k$ . When  $t_{k+1}$  and  $\hat{x}^k$  changed, we can still interpret  $x(\mu)$  as an inexact solution to the optimization problem (23) at dual signal  $(\mu, 0)$  for iteration  $k + 1$ . This gives rise to the usual rationale of an “upper”-oracle (see [19, Lemma 2.1] and [44]) since we are working in the concave dual. In practice, we just need to save past primal iterates with their true function values to set up an initial cutting plane model for  $\Theta_{k+1}$  or  $\Theta_{k+1}^R$  following Lemmas 1 and 3. This strategy proves highly effective, as the computational cost of maximizing the Lagrangian dual for a given  $k$  is highly reduced. This is illustrated in Figure 4.

### 3.4.3 Possible extensions to inexact oracles for $\Psi$

We finally note that our scheme can be combined with inexact computations of the slave problem, which is particularly useful if the set  $\mathcal{D}$  is hard to maximize over. Indeed, we have assumed so far that we can solve  $\sup_{d \in \mathcal{D}} \psi(x, d)$  for a given  $x$ ; in other words, that the oracle for  $\Psi$  is exact. As mentioned in the introduction, performing this maximization is not mandatory, since any  $d$  can provide us with valid linearizations for  $\Psi$  by (6). This inexactness (i.e., computing  $\Psi$  inexactly) can be handled by bundle methods. Following [44, Lemma 5.1], we need to append the test  $\delta^k < \frac{1}{2t_k} \|x^{k+1} - \hat{x}^k\|^2$  in between steps 2 and 3 of the algorithm. Whenever this condition holds true, the presence of noise is suspected and we suggest to apply the usual remedy [38] which consists in increasing  $t_k$  strictly and then return to Step 1 to produce a new iterate. The analysis of the effect of using inexact oracles of [44], reveals that only when nearing convergence, it is of importance to request some precision when computing  $\Psi$  to get a good final accuracy. Requesting more precision here means maximizing the mappings  $\psi(x, d)$  sufficiently well over  $\mathcal{D}$ . We refer the reader to [20, 54] for further details and alternatives. We note that in our numerical experiments computing  $\Psi$  is fast so that we have not explored this point any further.

## 4 Numerical experiments

The goal of the numerical experiments is twofold: first, illustrating that the approach of the paper can be deployed on a real-life application; second, comparing the cutting plane method versus the stabilized cutting plane method, showing the interests of the latter.

### 4.1 Experimental setting

Our numerical experiments are conducted on robust large scale unit-commitment problems as presented in section 2.1, similar to those that EDF, faces everyday. We briefly present here our unit-commitment model and data.

*Unit-commitment model and data* We use standard choices in the unit-commitment literature for unit modeling, i.e., a convex model for hydro valleys (e.g., [59]) and a standard non-convex model for thermal units (e.g., [25]). The overall model comes from a part of the set of French electricity generation assets.

Instance	date	# Thermal units	# Hydro valleys
1	15/01/2013	106	41
2	20/03/2013	106	32
3	13/05/2013	104	35
4	22/08/2013	104	34
5	25/10/2013	104	37
6	10/12/2013	104	40

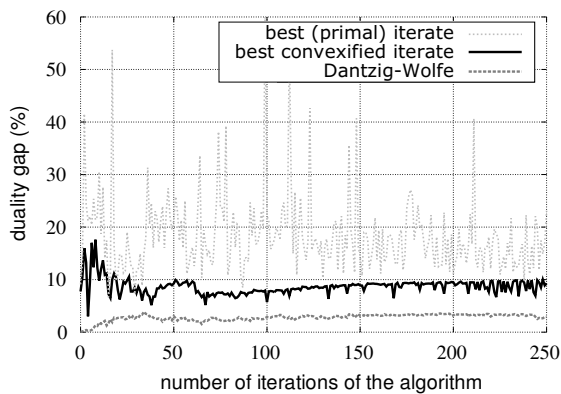
**Table 1** Instance label with corresponding date and numbers of units; the data sets contain 96 half hourly time steps.

The total number of variables describing the set  $X$  is around 50000 continuous variables, 27000 binary variables and we need around 815000 constraints. The data is taken from the operational data sets of 2013 of EDF: we have  $T = 96$  half hourly time steps, and 6 data sets for 6 days of 2013, given by Table 1. Moreover we generate a variant of each data set by perturbing the load data with a multiplicative factor along the lines of [60, Section 5.1] with three dispersion load scenarios (low, medium, and large). We hence obtain a total of 18 instances defined by its date and load factor.

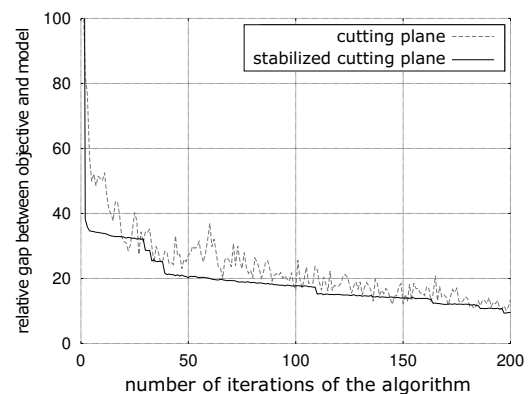
*Robust unit-commitment model* In this above-described large scale unit-commitment setting, we consider the robust 2-stage model of the form (3) as described in section 2.1. The coupling penalization function  $\Psi$  is defined from the piecewise affine functions  $\psi_t$  of (10) with EDF industrial choices for parameters  $a_{i,t}$  and  $b_{i,t}$  (see [23, Fig 2.2]).

For the model of uncertainty, our approach could handle any set  $\mathcal{D}$  for which we can easily solve (12). This includes in particular finite sets with small cardinality and Cartesian products of intervals. However these two models have a limited expressivity: on one hand the finite set is obviously limited; and on the other hand, the product of intervals is too large and does not express the dependencies between two successive time-steps that we have in our application. Then we choose the so-called “state-space represented” model as suggested in [43]. This model of uncertainty is finite but of very high cardinality, expressing temporal dependencies while preserving a very fast computability by dynamic programming. This model is illustrated and shortly described in appendix C; for a complete analysis, we refer to [43].

*Lagrangian approaches vs monolithic resolution approaches* In our context, the deterministic problem (i.e., without the robust penalization function  $\psi$  but with load-demand balance equations) is so large with the data instances described above, that it cannot be solved (to any decent precision) by a monolithic solver (e.g., CPLEX). On top of this, the state-space model for  $\mathcal{D}$  contains an exponential number of scenarios, in our case  $1.35 \times 10^{81}$ , so that an explicit reformulation involving all these scenarios is simply not possible. The Lagrangian dual approach is therefore the only viable alternative to solving the master problem. In general, as described in [50], Lagrangian dualization is one of the essential methods for solving large scale unit-commitment problems with extensive modeling detail. The Lagrangian approach produces lower-bounds and (infeasible) primal solutions of very good quality exploited by efficient heuristics. In related contexts using appropriate methods (e.g., [12, 24, 51]) solutions of very good quality, near 0.5 % optimality, can be obtained (e.g., [25]).



**Fig. 2** Comparison of the three generic heuristics on a typical run of the bundle method within the solving process



**Fig. 3** Illustration of the oscillation effect of the cutting plane algorithm versus the stabilized one on instance number 6



*Algorithmic choices* We use rudimentary choices of parameters after some preliminary tests. In particular we use a fixed prox-parameter for all data sets (which is theoretically justified, see the discussion in the appendix on bundle methods). We empirically chose the value  $t_k = 1000$  since it tends to “balance off” the terms in the objective function of problem (22). We also use the Dantzig-Wolfe heuristic (see section 3.4) as recovery step in the two algorithms, since it tends to produce much better feasible solutions. We illustrate this on a typical example in Figure 2.

## 4.2 Numerical results

We compare the stabilized cutting plane method of section 3.3 against the cutting plane method of section 3.2. All computations have been carried out using CPLEX 12.6 on an Intel Xeon X5670 westmere computer cluster with 8 Gb of reserved memory without using parallelization.

We consider large scale instances: the cost of the master problems are negligible in the total computing amount and the main computational cost in the solution process consists of solving the  $m$  subproblems (2). So we count the cost of the algorithms in terms of the total number of sub-problem resolutions, and the comparison is based on this cost.

Figure 3 shows a typical behaviour of the two algorithms on one instance. It immediately becomes apparent that the cutting plane method suffers significantly from oscillation and that the regularized variant shows a better behaviour as this aspect is concerned. Tables 2 and 3 report the complete comparison on all the instances with two different stopping thresholds. The value - corresponds to a computation that did not reach convergence within the time limit.

The tables clearly show that stabilization reduces the number of (outer-)iterations, and most importantly the number of oracle calls. Even instances that were not solved by the basic cutting plane method, can be solved by the stabilized method.

Instance	load factor	cp		stab. cp		improvement	
		# It.	Oracle Calls	# It.	Oracle Calls	# It.	Oracle Calls
15/01/2013	low	64	711	43	449	149%	158%
	medium	45	760	44	495	102%	154%
	high	41	536	26	317	158%	169%
20/03/2013	low	2045	2281	251	416	815%	548%
	medium	848	1082	157	317	540%	341%
	high	393	613	103	245	382%	250%
13/05/2013	low	3074	3862	439	964	700%	401%
	medium	1608	2404	266	720	605%	334%
	high	764	2315	319	974	239%	238%
22/08/2013	low	-	-	684	1098	$\infty\%$	$\infty\%$
	medium	-	-	662	1152	$\infty\%$	$\infty\%$
	high	2648	3452	356	790	744%	347%
25/10/2013	low	-	-	297	815	$\infty\%$	$\infty\%$
	medium	2923	3894	369	1044	792%	373%
	high	724	2163	209	1147	346%	189%
10/12/2013	low	236	916	96	398	246%	230%
	medium	136	661	55	321	247%	206%
	high	78	578	34	229	229%	252%
<b>average</b>		1042	1749	245	661	420 %	285 %

**Table 2** Comparison of the cutting plane method (column cp) and stabilized cutting plane method (column stab cp) for a stopping criteria  $\delta_{\text{stop}} = 5\%$ .

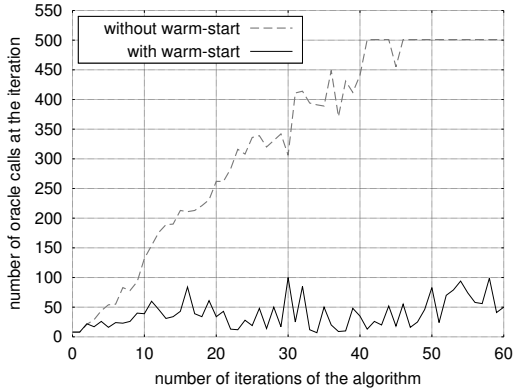
The impressive gain of the stabilized algorithm could be explained by two factors. First, the stabilized algorithm produces better iterates, which is the raison-d’être of this method, but not always easy to obtain in a context of mixed-integer programming ([66], where typically the method is not outperformed



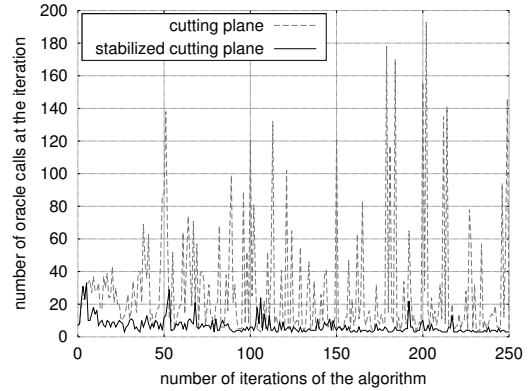
Instance	load factor	cp		stab. cp		improvement	
		# It.	Oracle Calls	# It.	Oracle Calls	# It.	Oracle Calls
15/01/2013	low	511	1163	218	901	234%	129%
	medium	253	1047	121	631	209%	166%
	high	203	895	100	605	203%	148%
20/03/2013	low	-	-	408	556	$\infty\%$	$\infty\%$
	medium	2146	2380	233	370	921%	643%
	high	1768	1988	299	445	591%	447%
13/05/2013	low	-	-	885	1438	$\infty\%$	$\infty\%$
	medium	-	-	538	1281	$\infty\%$	$\infty\%$
	high	-	-	728	1296	$\infty\%$	$\infty\%$
22/08/2013	low	-	-	1046	1499	$\infty\%$	$\infty\%$
	medium	-	-	657	1129	$\infty\%$	$\infty\%$
	high	-	-	1240	1696	$\infty\%$	$\infty\%$
25/10/2013	low	-	-	691	1321	$\infty\%$	$\infty\%$
	medium	-	-	708	1326	$\infty\%$	$\infty\%$
	high	-	-	494	1289	$\infty\%$	$\infty\%$
10/12/2013	low	3500	4505	360	940	972%	479%
	medium	1160	1895	317	775	366%	245%
	high	673	1442	193	644	349%	224%
average		1277	1914	513	1008	481%	310%

**Table 3** Comparison of the cutting plane method (column cp) and stabilized cutting plane method (column stab cp) for a stopping criteria  $\delta_{\text{stop}} = 1\%$ .

uniformly on all instances). Second, the strategy of the warm-starting presented in section 3.4, though highly effective for both algorithms, is even more efficient for the stabilized algorithm. We illustrate the effect of warm-starting on a typical example on Figures 4 and 5.



**Fig. 4** Comparison of the number of inner iterations (within Step 1) for each outer iteration, with and without warm-starting



**Fig. 5** Less subproblem resolutions for quadratic algorithm using warm-start

We see on Figure 4 that the computational cost of maximizing the Lagrangian dual for a given  $k$  is highly reduced for the typical example. Figure 5 then illustrates that the warm start is more efficient for the stabilized algorithm which, roughly speaking, explores the solutions in a more continuous way. The shown efficiency of warm-starting (under regularization) seem to be observations of some sort of general mechanism. Indeed, the phenomenon was already reported in [52] but also in [66] (less feasibility cuts).

## 5 Conclusions

We have proposed and studied a generic decomposition scheme for robust block-structured problems. The main algorithm is a regularized cutting-plane method intertwining two layers of bundle methods that scales up to large instances. Its practical performance relies on the combination of efficient numerical features, including warm-starting and heuristics. Its theoretical analysis is not covered by existing results because of the technical difficulty of the inexact resolution of the master problem. We have provided a convergence analysis using basic arguments. We have illustrated the approach with numerical experiments on large scale instances coming from unit-commitment.

Though cutting-plane methods are basic methods in robust optimization, regularized cutting-plane algorithms are not often presented or studied in this context; in contrast, it is sometimes reported to be not useful. Our numerical experiments show clearly that for the robust block-structured optimization problems we consider here, the decomposition scheme is much more efficient with regularization, in terms of computing times and stability. We also underline that the numerical features of the two cutting-plane algorithms are essentially the same, and we advocate that the jump from the standard to the regularized one is not complicated, neither conceptually nor implementation-wise. The main contribution of this paper is thus to promote the combination of cutting-plane and regularization in the context of robust optimization. To support this message we release<sup>1</sup> the code implementing the regularized algorithm.

**Acknowledgements** Part of this work was done during the second author’s internship at EDF R&D. The first and the third author gratefully acknowledge the support of PGMO project “advanced nonsmooth optimization methods for stochastic programming”.

## References

1. C. Bandi and D. Bertsimas. Tractable stochastic analysis in high dimensions via robust optimization. *Mathematical Programming*, 134(1):23–70, 2012.
2. H. Ben Amor, J. Desrosiers, and A. Frangioni. On the Choice of Explicit Stabilizing Terms in Column Generation. *Discrete Applied Mathematics*, 157(6):1167–1184, 2009.
3. S. Ben-Salem. *Gestion Robuste de la production électrique à horizon court-terme*. PhD thesis, Ecole Centrale Paris, Mars 2011.
4. A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton University Press, 2009.
5. A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, November 1998.
6. A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13, August 1999.
7. J.F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.
8. D. Bertsimas, D. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.
9. D. Bertsimas, I. Dunning, and M. Lubin. Reformulation versus cutting-planes for robust optimization. *Computational Management Science*, 13(2):195–217, 2016.
10. D. Bertsimas, E. Litvinov, X. A. Sun, J. Zhao, and T. Zheng. Adaptive robust optimization for the security constrained unit commitment problem. *IEEE Transactions on Power Systems*, 28(1):52–63, March 2013.
11. D. Bertsimas and M. Sim. Tractable approximations to robust conic optimization problems. *Mathematical Programming*, 107(1):5–36, 2006.
12. A. Borghetti, A. Frangioni, F. Lacalandra, and C.A. Nucci. Lagrangian heuristics based on disaggregated bundle methods for hydrothermal unit commitment. *IEEE Transactions on Power Systems*, 18:313–323, 2003.
13. O. Briant, C. Lemaréchal, Ph. Meurdesoif, S. Michel, N. Perrot, and F. Vanderbeck. Comparison of bundle and classical column generation. *Mathematical Programming*, 113(2):299–344, 2008.
14. C. C. Caroe and J. Tind. L-shaped decomposition of two-stage stochastic programs with integer recourse. *Math. Programming*, 83:451–464, 1998.
15. N. Chatzipanagiotis, D. Dentcheva, and M. M. Zavlanos. An augmented lagrangian method for distributed optimization. *Mathematical Programming*, 152(1):405–434, 2015.
16. F.H. Clarke. *Optimisation and Nonsmooth Analysis*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1987.

<sup>1</sup> <https://github.com/NLebbe/rbsopb>

17. G. Codato and M. Fischetti. Combinatorial benders' cuts for mixed-integer linear programming. *Operations Research*, 54(4):756–766, 2006.
18. R. Correa and C. Lemaréchal. Convergence of some algorithms for convex minimization. *Mathematical Programming*, 62(2):261–275, 1993.
19. W. de Oliveira and J. Eckstein. A bundle method for exploiting additive structure in difficult optimization problems. Preprint [http://www.optimization-online.org/DB\\_HTML/2015/05/4935.html](http://www.optimization-online.org/DB_HTML/2015/05/4935.html), pages 1–18, 2015.
20. W. de Oliveira and C. Sagastizábal. Level bundle methods for oracles with on demand accuracy. *Optimization Methods and Software*, 29(6):1180–1209, 2014.
21. W. de Oliveira, C.A. Sagastizábal, and S. Scheimberg. Inexact bundle methods for two-stage stochastic programming. *SIAM Journal on Optimization*, 21(2):517–544, 2011.
22. J. Van Dinter, S. Rebenack, J. Kallrath, P. Denholm, and A. Newman. The unit commitment model with concave emissions costs: a hybrid benders decomposition with nonconvex master problems. *Annals of Operations Research*, 210(1):361–386, 2013.
23. L. Dubost, R. Gonzalez, and C. Lemaréchal. A primal-proximal heuristic applied to french unitcommitment problem. *Mathematical Programming*, 104(1):129–151, 2005.
24. S. Feltenmark and K.C. Kiwiel. Dual applications of proximal bundle methods, including lagrangian relaxation of nonconvex problems. *SIAM Journal on Optimization*, 10(3):697–721, 2000.
25. A. Frangioni, C. Gentile, and F. Lacalandra. Sequential Lagrangian-MILP Approaches for Unit Commitment Problems. *International Journal of Electrical Power and Energy Systems*, 33:585–593, 2011.
26. A. M. Geoffrion. Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.
27. L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064, July 2006.
28. L. El Ghaoui, F. Oustry, and H. Lebret. Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization*, 9(1):33–52, 1998.
29. S. Guo, H. Xu, and L. Zhang. Stability analysis for mathematical programs with distributionally robust chance constraint. *SIAM Journal on Optimization (to appear)*, pages 1–27, 2017.
30. W. Hare, C. Sagastizábal, and M. Solodov. A proximal bundle method for nonconvex functions with inexact oracles. *Computational Optimization and Applications*, 63(1):1–28, 2016.
31. H. Heitsch and W. Römis. A note on scenario reduction for two-stage stochastic programs. *Operations Research Letters*, 35(6):731–738, 2007.
32. J.B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I*. Number 305 in Grundlehren der mathematischen Wissenschaften. Springer-Verlag Berlin Heidelberg, 2nd edition, 1996.
33. J.B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II*. Number 306 in Grundlehren der mathematischen Wissenschaften. Springer-Verlag Berlin Heidelberg, 2nd edition, 1996.
34. J. N. Hooker and G. Ottosson. Logic-based benders decomposition. *Math. Programming*, 96:33–60, 2003.
35. J.E. Kelley. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.
36. K.C. Kiwiel. A method for solving certain quadratic programming problems arising in non-smooth optimization. *IMA journal of numerical Analysis*, 6(2):137–152, 1986.
37. K.C. Kiwiel. A cholesky dual method for proximal piecewise linear programming. *Numerische Mathematik*, 68:325–340, 1994.
38. K.C. Kiwiel. A proximal bundle method with approximate subgradient linearizations. *SIAM Journal on Optimization*, 16(4):1007–1023, 2006.
39. A. Kolokolov and N. Kosarev. Analysis of decomposition algorithms with benders cuts for  $p$ -median problem. *Journal of Mathematical Modelling and Algorithms*, 5(2):189–199, 2006.
40. C. Lemaréchal. An extension of davidon methods to nondifferentiable problems. *Mathematical programming study*, 3:95–109, 1975.
41. C. Lemaréchal, A. Nemirovskii, and Y. Nesterov. New variants of bundle methods. *Math. Programming*, 69(1):111–147, 1995.
42. J. Malick, W. de Oliveira, and S. Zaourar. Uncontrolled inexact information within bundle methods. *EURO Journal of Computational Optimization*, 5(1):5–29, 2017.
43. M. Minoux. Two-stage robust optimization, state-space representable uncertainty and applications. *RAIRO-Operations Research*, 48:455–475, 2014.
44. W. de Oliveira, C. Sagastizábal, and C. Lemaréchal. Convex proximal bundle methods in depth: a unified analysis for inexact oracles. *Math. Prog. Series B*, 148:241–277, 2014.
45. G. Pagès and J. Printems. Optimal quadratic quantization for numerics: the gaussian case. *Monte Carlo Methods and Applications*, 9(2):135–166, 2003.
46. R.T. Rockafellar and R. J.-B. Wets. *Variational Analysis*, volume 317 of *Grundlehren der mathematischen Wissenschaften*. Springer Verlag Berlin, 3rd edition, 2009.
47. G. K. D. Sahiridis, M. Minoux, and M. G. Ierapetritou. Accelerating benders method using covering cut bundle generation. *International Transactions In Operational Research*, 17:221–237, 2010.
48. H.D. Sherali and B. J. Lunday. On generating maximal nondominated benders cuts. *Annals of Operations Research*, 210(1):57–72, 2013.
49. Y. Song and J. Luedtke. An adaptive partition-based approach for solving two-stage stochastic programs with fixed recourse. *SIAM Journal on Optimization*, 25(3):1344–1367, 2015.

50. M. Tahanan, W. van Ackooij, A. Frangioni, and F. Lacalandra. Large-scale unit commitment under uncertainty: a literature survey. *4OR*, 13(2):115–171, 2015.
51. S. Takriti and J.R. Birge. Using integer programming to refine lagrangian-based unit commitment solutions. *IEEE Transactions on Power Systems*, 15(1):151–156, 2000.
52. W. van Ackooij. Decomposition approaches for block-structured chance-constrained programs with application to hydro-thermal unit commitment. *Mathematical Methods of Operations Research*, 80(3):227–253, 2014.
53. W. van Ackooij. A comparison of four approaches from stochastic programming for large-scale unit-commitment. *EURO Journal on Computational Optimization*, 5(1):119–147, 2017.
54. W. van Ackooij, V. Berge, W. de Oliveira, and C. Sagastizábal. Probabilistic optimization via approximate p-efficient points and bundle methods. *Computers & Operations Research*, 77:177–193, 2017.
55. W. van Ackooij and W. de Oliveira. Level bundle methods for constrained convex optimization with various oracles. *Computation Optimization and Applications*, 57(3):555–597, 2014.
56. W. van Ackooij, W. de Oliveira, and Y. Song. An adaptive partition-based level decomposition for solving two-stage stochastic programs with fixed recourse. *Submitted draft, preprint available [http://www.optimization-online.org/DB\\_HTML/2016/03/5373.html](http://www.optimization-online.org/DB_HTML/2016/03/5373.html)*, pages 1–18, 2016.
57. W. van Ackooij and A. Frangioni. Incremental bundle methods using upper models. *Submitted paper; Technical report : <http://eprints.adm.unipi.it/2357/>*, pages 1–25, 2016.
58. W. van Ackooij, A. Frangioni, and W. de Oliveira. Inexact stabilized Benders’ decomposition approaches: with application to chance-constrained problems with finite support. *Computational Optimization And Applications*, 65(3):637669, 2016.
59. W. van Ackooij, R. Henrion, A. Möller, and R. Zorgati. Joint chance constrained programming for hydro reservoir management. *Optimization and Engineering*, 15:509–531, 2014.
60. W. van Ackooij and J. Malick. Decomposition algorithm for large-scale two-stage unit-commitment. *Annals of Operations Research*, 238(1):587–613, 2016.
61. W. van Ackooij and C. Sagastizábal. Constrained bundle methods for upper inexact oracles with application to joint chance constrained energy problems. *SIAM Journal on Optimization*, 24(2):733–765, 2014.
62. R.M. van Slyke and R.J.B. Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal of Applied Mathematics*, 17:638–663, 1969.
63. P. Wentges. Accelerating benders’ decomposition for the capacitated facility location problem. *Mathematical Methods of Operations Research*, 44(2):267–290, 1996.
64. Y. Yang and J. M. Lee. A tighter cut generation strategy for acceleration of benders decomposition. *Computers and Chemical Engineering*, 44:84–93, 2012.
65. G. Zakeri, A. Philpott, and D. M. Ryan. Inexact cuts in benders decomposition. *SIAM Journal on Optimization*, 10(3):643–657, 2000.
66. S. Zaourar. *Optimisation convexe non-différentiable et méthodes de décomposition en recherche opérationnelle*. PhD thesis, University of Grenoble, November 2014.
67. S. Zaourar and J. Malick. Quadratic stabilization of benders decomposition. pages 1–22, 2014. Draft submitted; Privately communicated.
68. B. Zeng and L. Zhao. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operation Research Letters*, 41(5):457–461, 2013.
69. V. Zverovich, C. I. Fábán, E. F. D. Ellison, and G. Mitra. A computational study of a solver system for processing two-stage stochastic lps with enhanced benders decomposition. *Mathematical Programming Computation*, 4(3):211–238, 2012.

## A Brief overview of bundle methods in nonsmooth optimization

This appendix provides a brief description of bundle methods using standard notation and terminology, which are also the ones of this paper. For an in-depth description, we refer to the textbook [33] and the recent article [44].

We are interested in solving  $\min_x f(x)$ , where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is a nonsmooth convex mapping known through an ”oracle” returning, for a given  $x \in \mathbb{R}^n$  as an input, the value  $f(x)$  and a subgradient  $g \in \partial f(x)$  (or approximations of them). Given a set of iterates  $x^1, \dots, x^k$  for which the oracle has been called, we can set up the cutting plane model  $\tilde{f}_k$  for  $f$  by defining

$$\tilde{f}_k(x) := \max_{\ell=1, \dots, k} \left\{ f(x^\ell) + \langle g^\ell, x - x^\ell \rangle \right\},$$

where  $g^\ell \in \partial f(x^\ell)$ . The cutting-plane algorithm, which can be traced back to [35], consists of defining the next iterate by solving  $\min_x \tilde{f}_k(x)$ , which can be formulated as a linear problem.

The cutting-plane method suffers from various drawbacks (including stability and memory issues) and an improved scheme is available: regularized cutting-plane methods or so-called bundle methods, dating back to the seminal work of [40]. These methods use a quadratic stabilization to ensure that the next iterate is not far from the best current iterate, called stability center  $\hat{x}$ . The next iterate is the solution of

$$\min_x \tilde{f}_k(x) + \frac{1}{2t_k} \|x - \hat{x}^k\|^2, \quad (29)$$

which is a convex quadratic program (easy to solve by specialized methods [36, 37]). The predicted decrease  $\delta_k := f(\hat{x}^k) - \hat{f}_k(x^{k+1})$  is then used to update the stability center: if enough decrease is observed, i.e. the inequality

$$f(x^{k+1}) \leq f(\hat{x}^k) - m\delta_k. \quad (30)$$

holds, the new iterate  $x^{k+1}$  becomes the new stability center  $\hat{x}^{k+1}$ ; Otherwise the information from the oracle is just used to improve the cutting plane model. From a variational analysis perspective, bundle methods are thus damped proximal algorithms where the inner proximal iterations are stopped whenever sufficient decrease is observed on  $f$ .

Note that the proximal parameter  $t_k$  is basically free: for exact oracles, convergence is guaranteed if  $t_k$  stays within bounds (see [18]). In practice,  $t_k$  can be arbitrary if the stability center was changed;  $t_k$  should not increase if the stability center was not changed;  $t_k$  should strictly increase if the oracle is subject to imprecise calculations during noise detection steps (see e.g., [61, eq. (2.16)]). A complete description of asymptotical rules can be found in [57].

On top of stabilizing iterates, bundle methods also come up with the possibility of aggregating cutting-plane information into an aggregate cut. In theory, we can delete all but two cutting planes (the aggregate cut and the cut belonging to the current stability center).

## B Benders decomposition: terminology and references

The so-called Benders decomposition hinges on the observation that fixing some variables makes certain problems much easier. This observation, originally made for problems with an underlying linear structure [7] potentially with stochastic aspects [62], was generalized to problems with some underlying convexity in the seminal work [26]. We recall in this appendix the terminology and some important references about this classical method in operation research in a broad sense.

The slave problem consists in the resulting problem with the fixed variables. The master problem is related to finding the optimal allocation of the previously fixed variables. Since the domain of the value function need not be the whole space, the master problem is augmented with the so called *feasibility cuts* (i.e., an outer approximation of the convex domain of the value function). The master problem is also enriched with *optimality cuts*, as a matter of fact, a cutting-plane model of the value function. Benders algorithm is thus as a variant of the cutting-plane method, presented previously, and therefore subject to the well-known oscillation effect and slow convergence. Regularized Benders algorithms have been recently proposed [67] and [58].

Many articles, including [22, 47, 48, 63, 64] concretely deal with strategies for generating good (optimality) cuts, occasionally with a problem-dependent flavour. The authors of [17] are concerned with generating strong feasibility cuts and provide an important contribution in this view. Further improvements to the general scheme are concerned with a relaxation of the convexity assumption of the slave problem by allowing for integer variables [14] or the use of generalized “logic” duality (called inference duality) in [34]. In [65], it is moreover suggested to inexactly solve the slave problem to compute some inexact but “cheap” cuts. In general, an appropriate choice of cuts (approximating the value function) is crucial, as illustrated in [39].

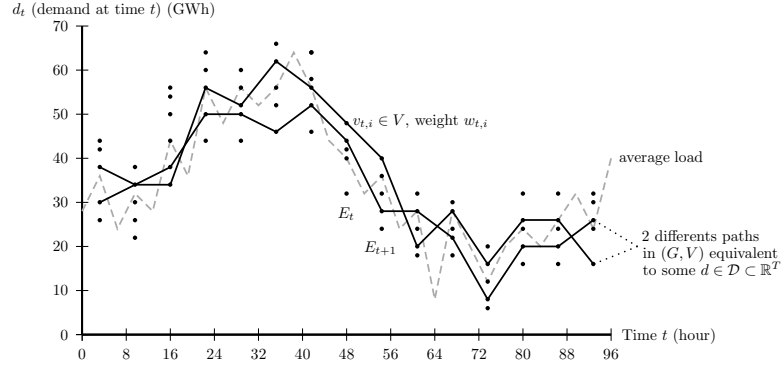
Let us finally illustrate optimality and feasibility cuts more specifically on the following parametrized linear program:

$$\psi(x) := \min_y \{q^\top y : Wy \leq b - Tx\}.$$

The inequality system  $Wy \leq b - Tx$  need not admit a solution for all  $x$  and that consequently  $\psi(x) = \infty$  may happen for some  $x$ . Minimizing the convex mapping  $\psi$  then amounts to handling also the hidden constraints  $x \in \text{dom } \psi$ . By strong duality, we have  $\psi(x) = \max_{\lambda \geq 0} \{\lambda^\top (b - Tx) : W^\top \lambda \geq q\}$ , so that any optimal dual multiplier  $\lambda^*$  satisfies  $-T^\top \lambda^* \in \partial \psi(x)$ . When  $Wy \leq b - Tx$  does not admit a feasible solution, by Farkas Lemma an unbounded ray can be identified and used to define an outer linearization for  $\text{dom } \psi$ . Note finally that, in the general non-linear situation, first order information about  $\psi$  can also be identified with tools from nonsmooth analysis (e.g., [46, Theorem 10.13]), but feasibility cuts will need to be derived by solving an auxiliary problem typically involving the minimization of slack variables.

## C A rich but computationally cheap model of uncertainty

For our numerical experiments, we choose to use the uncertainty set  $\mathcal{D}$  introduced by [43]. This so-called “state-space represented” model has a rich expressivity while being computationally easy to manipulate. This model has an exponential number of scenarios expressing temporal dependencies, while preserving a very fast computability by basic dynamic programming. We refer to [43] more details and a complete analysis. This model is illustrated in Figure 6. Yet, for the convenience of the reader, we describe here the main ideas briefly: At each time step  $t \in \tau$ , we set up a set of nodes  $E_t$  containing both a value for  $d_t$  and a weight  $w_t$ . This set of nodes is assumed to be sorted according to increasing values for  $d_t$ . Using this set of nodes we set up a graph  $(G, V)$ , with  $G = \bigcup_{t \in \tau} E_t$ , such that  $V$  connects all nodes in  $E_t$  to those in  $E_{t+1}$  that are not further apart than  $H$ , i.e., node  $i \in E_t$  is connected to node  $j \in E_{t+1}$  if and only if their local labels  $|\mathcal{L}(i) - \mathcal{L}(j)| \leq H$ . At least one of the nodes in  $E_t$  is assumed to have a zero weight and  $H$  is assumed to be such that each



**Fig. 6** Illustration of the uncertainty model used in the numerical experiments

node is connected to the node of zero weight. The set  $\mathcal{D}$  is now given by all paths in  $(G, V)$  satisfying  $\sum_{t \in \mathcal{T}} w_t \leq W_{\max}$  for a given maximum budget of uncertainty  $W_{\max}$ . As a consequence,  $|\mathcal{D}|$  is huge, but computing the sup over  $\mathcal{D}$  amounts to applying a simple 1-dimensional dynamic programming principle.